Modern Web development with SAP (Hands on) Vue.js + axios

beyhan_meyrali
Active Contributor

2019 Feb 04 12:49 PM

6 Kudos

6,514 Views

Hi,

Update 13.12.2022 -> https://blogs.sap.com/2022/12/12/how-to-create-a-vue.js-app-with-vs-code-and-and-deploy-to-sap-netwe...

if you are living in SAP world then you may think that i meant of Fiori. But if you are a developer, familiar with different development environments then you may know, web world is evolving much faster than what SAP can provide to developers. Luckily I work in a R&D department and we have a team of .Net Developers who are using cutting edge tools for their projects. Thanks to them, now i use Vue as framework to develop UIs and rest api instead of OData.

Creating OData services on Gateway and then trying achieve something with limited abilities of Fiori framework requires a lots of patience and a lots of time. And I think SAP should let developers to use any tool they want to develop WEB UI with rest api, rather than pushing them to use OData services and Fiori framework. Therefore i have created an alternative way of creating responsive and reactive web applications.In that way we can create great web applications with any framework we like or simply a rest based web api to share information instead of classical web services.

Now, Lets create a web api for ourselves and use it with **VueJs** and **axios**. Following parts of the article will show you how to create **a BSP page which acts as rest based web api** and how to create a very basic json consuming page. Web api result can be returned as Json, Json Xml or as xml.

Overview;

We will create a BSP application, application will have a controller to process requests and a view for returning Json from controller. And

finally another view, just to consume web api. That is all.

Codes and structure are below. In order to understand example better to create a BSP and try it yourself. I am sure it will **inspire** you to create modern reactive/responsive web applications on SAP with rest api.

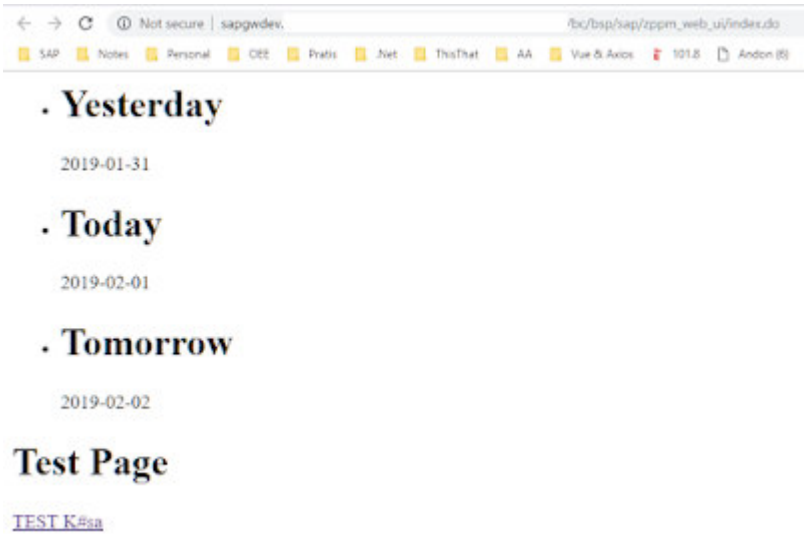
You can download Vue.js and axios from links below.

https://cdn.jsdelivr.net/npm/vue/dist/vue.js

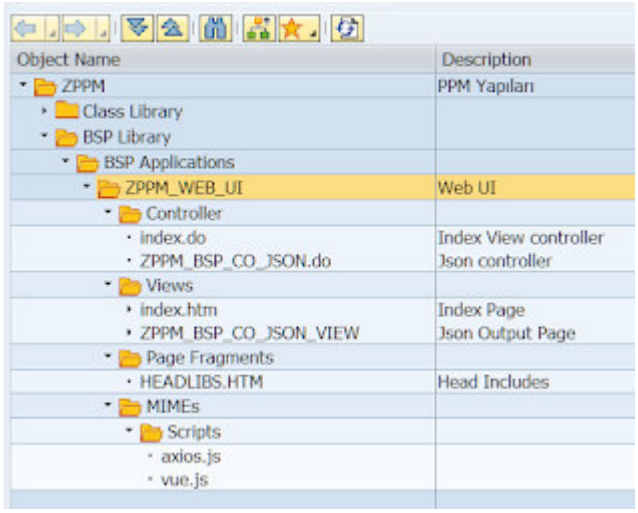
https://unpkg.com/axios/dist/axios.min.js

Note: Following example is used in PPM package therefore initials are zppm... You can name them anything you like. Some classes are base for my own bsp framework to handle session and request params. You can use same logic if you like.

Result will be;

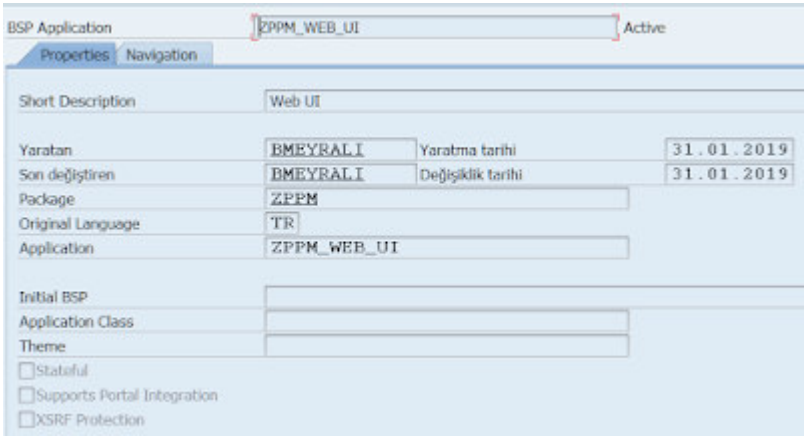


Application Structure;



1- Lets create a BSP application.

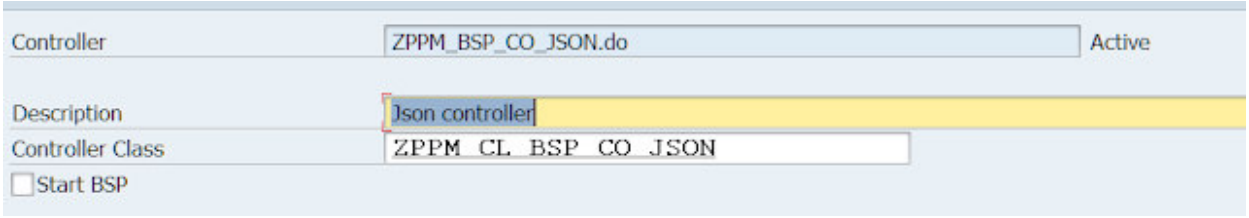
"ZPPM_WEB_UI"



2- Create a controller and class for controller.

Controller : "ZPPM_BSP_CO_JSON.do"

Controller Class: "ZPPM_CL_BSP_CO_JSON"



Code of Controller and Model class below...

3- Create a view **to act as web api**, which will show json data.

"ZPPM_BSP_CO_JSON_VIEW"

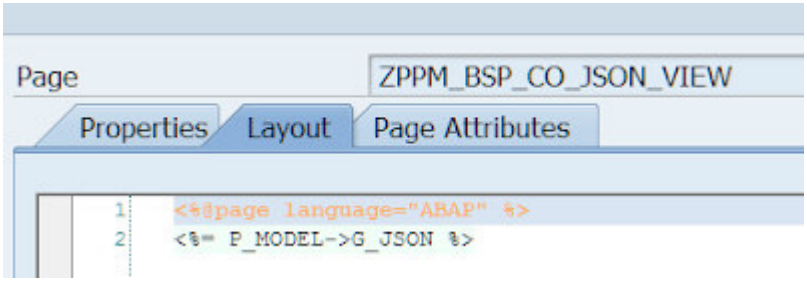
Mime type of the view is : application/json; charset=utf-8

Content of View(No html code at all!):

```
<%@page language="ABAP" %>
<%= P_MODEL->G_JSON %>
```

Page attributes:
P_CONTROLLER_NAME TYPE STRING

P_MODEL TYPE REF TO ZPPM_CL_BSP_CO_JSON_MO "Model object contains json data which is retrieved by axios and displayed with vue on index.htm



4- Lets create a page to make use of web api. Page contains some dummy data and code for vue.
View name : "index.htm"
Controller : index.do
Controller Class: ZPPM_CL_BSP_CO_INDEX "Code below
Content of view :

```
<HTML>
<HEAD>
  <!-- production version, optimized for size and speed -->
  <script src="Scripts/vue.js"></script>
  <script src="Scripts/axios.js"></script>
</HEAD>

<body>

  <div id="app2">
    <div v-if="loading">
      loading...
    </div>
    <div v-else>
      <ul>
        <li v-for="item in items.ITAB">
          <h1>
            {{item.DAY}}
          </h1>
          <p>
            {{item.DATE}}
          </p>
        </li>
      </ul>
    </div>
  </div>


  <h1>Test Page</h1>
  <p><a href="ZPPM_BSP_CO_JSON.do">TEST</a></p>

  <script>

    const URL = 'ZPPM_BSP_CO_JSON.do';
    var app2 = new Vue({
      el: '#app2',
      data() {
        return {
          loading: true,
      posts: [] // add posts here so reactivity is working, also unde
    }
  },
```

```
  created() {
    //this.loading = true --> not needed already set in data
    //op:03 represents Json format, see base class constants: ZBSP_CL
    axios.post(URL, {
      params: {
        op: '03'
      }}).then((response) => {
      // console.log(response.data, this)
      this.items = response.data
      this.loading = false
    })
    }
  })
</script>
</body>

</HTML>
```

Rest of the page contains classes of controller and base classes.

Json controller class. Pass request parameters to your model classes, process data and return result as Json. This Json is returned as data in axios response and shown on index page with Vue.

```
CLASS ZPPM_CL_BSP_CO_JSON DEFINITION
PUBLIC
  "Kind of framework to be used in feature BSP pages
  INHERITING FROM ZBSP_CL_BASE_CONTROLLER
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.

    "Page Data Model
    DATA P_MODEL TYPE REF TO ZPPM_CL_BSP_CO_JSON_MO .   "Class that c

    "Initialize Class
    METHODS DO_INIT REDEFINITION .
    METHODS DO_REQUEST REDEFINITION .

  PROTECTED SECTION.

    "Creates Page Model BO
    METHODS CREATE_PAGE_MODEL .

    "Set contoller params
    METHODS SET_CONTROLLER_PARAMS REDEFINITION .

  PRIVATE SECTION.
    DATA: GT_REQ_PRMS TYPE TIHTTPNVP.

    "Refresh Page Data
    METHODS REFRESH_PAGE_DATA.

    "Process Request
    METHODS PROCESS_REQUEST.

    "View Page
    METHODS VIEW_PAGE.

ENDCLASS.



CLASS ZPPM_CL_BSP_CO_JSON IMPLEMENTATION.
```

```
* <SIGNATURE>---------------------------------------------------------
* | Instance Protected Method ZPPM_CL_BSP_CO_JSON->CREATE_PAGE_MODEL
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD CREATE_PAGE_MODEL.

    "Create pages model class
    ME->P_MODEL ?= ME->CREATE_MODEL(
      CLASS_NAME = C_PAGE_MODEL_CLS
      MODEL_ID   = C_PAGE_MODEL_ID ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_JSON->DO_INIT
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD DO_INIT.
    CALL METHOD SUPER->DO_INIT.

    SET_CONTROLLER_PARAMS( ).
    CREATE_PAGE_MODEL( ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_JSON->DO_REQUEST
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD DO_REQUEST.
    CALL METHOD SUPER->DO_REQUEST.

    "Lets store parameters so we can make use of them in controller
    REFRESH GT_REQ_PRMS.
    RUNTIME->SERVER->REQUEST->GET_FORM_FIELDS( CHANGING FIELDS = GT_R

    PROCESS_REQUEST( ).

    VIEW_PAGE( ).
  ENDMETHOD.
```

```
* <SIGNATURE>---------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_JSON->PROCESS_REQUEST
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD PROCESS_REQUEST.
    "You can implement a controller logic here
    "Get Post/Get Parameters and call model according those parameter
    "We can return Json, Json xml or xml. We will use that variable t
    CLEAR P_MODEL->GV_OUT_TYPE.

    READ TABLE GT_REQ_PRMS ASSIGNING FIELD-SYMBOL(<WAP>)
    WITH KEY NAME = P_MODEL->C_OUT_PRM_NAME.

    IF SY-SUBRC IS INITIAL.
      P_MODEL->GV_OUT_TYPE = <WAP>-VALUE.
    ENDIF.

    "Process Request
    REFRESH_PAGE_DATA( ).
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_JSON->REFRESH_PAGE_DATA
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD REFRESH_PAGE_DATA.

    "Get Data
    "No logic is implemented. that will return just a dummy data
    ME->P_MODEL->GET_DATA( ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Protected Method ZPPM_CL_BSP_CO_JSON->SET_CONTROLLER_PAR
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD SET_CONTROLLER_PARAMS.
```

```
    C_PAGE                      = 'ZPPM_BSP_CO_JSON_VIEW'.
    C_PAGE_MODEL_CLS            = 'ZPPM_CL_BSP_CO_JSON_MO'.
    C_PAGE_MODEL_ID             = 'ZPPM_CL_BSP_CO_JSON_MO'.
    C_PAGE_MODEL_VAR            = 'P_MODEL'.
    C_PAGE_CONTROLLER_NAME_VAR = 'P_CONTROLLER_NAME'.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_JSON->VIEW_PAGE
* +-------------------------------------------------------------------
* +-------------------------------------------------------------------
  METHOD VIEW_PAGE.
    "View Page
    DATA: LV_VIEW TYPE REF TO IF_BSP_PAGE.
    LV_VIEW = CREATE_VIEW( VIEW_NAME = C_PAGE ).

    "We will pass json data as part of page attributes object
    LV_VIEW->SET_ATTRIBUTE( NAME  = C_PAGE_MODEL_VAR VALUE = ME->P_MC

    LV_VIEW->SET_ATTRIBUTE( NAME  = C_PAGE_CONTROLLER_NAME_VAR VALUE

    CALL_VIEW( LV_VIEW ).
  ENDMETHOD.
ENDCLASS.
```

My Base Contoller class to handle requests and session variables. Power of OOP 🙂

```
CLASS ZBSP_CL_BASE_CONTROLLER DEFINITION
  PUBLIC
  INHERITING FROM CL_BSP_CONTROLLER2
  ABSTRACT
  CREATE PUBLIC .

  PUBLIC SECTION.

    "Controller Params
    DATA PARAMS TYPE ZBSP_TY_S_CONTROLLER_PARAMS .
    DATA GV_DURUM TYPE ZUTIL_S_DURUM .

    "Set Session Variable
    METHODS SET_SESSION_VAR
      IMPORTING
        VALUE(I_NAME)  TYPE ZBSP_E_STR
        VALUE(I_VALUE) TYPE ANY
      EXPORTING
        VALUE(E_DURUM) TYPE ZUTIL_S_DURUM .

    "Get Session Variable
    METHODS GET_SESSION_VAR
      IMPORTING
        VALUE(I_NAME)  TYPE ZBSP_E_STR
      EXPORTING
        VALUE(E_DURUM) TYPE ZUTIL_S_DURUM
        VALUE(E_VALUE) TYPE ANY .

    "On Request process, get url params
    METHODS DO_REQUEST
         REDEFINITION .

    METHODS DO_INIT
         REDEFINITION .

  PROTECTED SECTION.

    "Page and model names
    DATA C_PAGE TYPE STRING VALUE '' ##NO_TEXT.
    DATA C_PAGE_MODEL_CLS TYPE SEOCLSNAME VALUE '' ##NO_TEXT.
    DATA C_PAGE_MODEL_ID TYPE STRING VALUE '' ##NO_TEXT.
    DATA C_PAGE_MODEL_VAR TYPE STRING VALUE 'P_MODEL' ##NO_TEXT.
```

```
    DATA C_PAGE_CONTROLLER_NAME_VAR TYPE STRING VALUE 'P_CONTROLLER_N
    "Populated in DO_REQUEST
    DATA GT_FORM_FIELDS TYPE TIHTTPNVP .

    "Set Controller Params for a view
    METHODS SET_CONTROLLER_PARAMS
         ABSTRACT .

  PRIVATE SECTION.

    "Gets Form Fields
    METHODS FILL_FORM_FIELDS.

ENDCLASS.



CLASS ZBSP_CL_BASE_CONTROLLER IMPLEMENTATION.


* <SIGNATURE>-------------------------------------------------------
* | Instance Public Method ZBSP_CL_BASE_CONTROLLER->DO_INIT
* +-------------------------------------------------------------------
* +-------------------------------------------------------------------
  METHOD DO_INIT.
    CALL METHOD SUPER->DO_INIT.
    RESPONSE->SET_HEADER_FIELD( NAME = 'Expires' VALUE = 'Thu, 01 Mar

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------
* | Instance Public Method ZBSP_CL_BASE_CONTROLLER->DO_REQUEST
* +-------------------------------------------------------------------
* +-------------------------------------------------------------------
  METHOD DO_REQUEST.
    CALL METHOD SUPER->DO_REQUEST.

    FILL_FORM_FIELDS( ).
  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------
```

```
* | Instance Private Method ZBSP_CL_BASE_CONTROLLER->FILL_FORM_FIELDS
* +----------------------------------------------------------------------
* +----------------------------------------------------------------------
  METHOD FILL_FORM_FIELDS.
    "Fill GT_FORM_FIELDS from URL Parameters
    DATA: EXREF TYPE REF TO CX_ROOT.
    TRY .

        "Get the URL variables and save them into the table.
        REFRESH GT_FORM_FIELDS.
        RUNTIME->SERVER->REQUEST->GET_FORM_FIELDS( CHANGING FIELDS =

        FIELD-SYMBOLS: <WA> TYPE LINE OF TIHTTPNVP.
        LOOP AT GT_FORM_FIELDS ASSIGNING <WA>.
          TRANSLATE <WA>-NAME TO UPPER CASE.
        ENDLOOP.

      CATCH CX_ROOT INTO EXREF.
    ENDTRY.
  ENDMETHOD.


* <SIGNATURE>----------------------------------------------------------
* | Instance Public Method ZBSP_CL_BASE_CONTROLLER->GET_SESSION_VAR
* +----------------------------------------------------------------------
* | [--->] I_NAME                         TYPE        ZBSP_E_STR
* | [<---] E_DURUM                        TYPE        ZUTIL_S_DURUM
* | [<---] E_VALUE                        TYPE        ANY
* +----------------------------------------------------------------------
  METHOD GET_SESSION_VAR.
    DATA: EXREF TYPE REF TO CX_ROOT.
    TRY .
        RUNTIME->SERVER->SET_SESSION_STATEFUL(  ).

        DATA: LV_COOKIE_NAME TYPE ZBSP_E_STR.
        LV_COOKIE_NAME = RUNTIME->APPLICATION_NAME && I_NAME.

        CALL METHOD CL_BSP_SERVER_SIDE_COOKIE=>GET_SERVER_COOKIE
          EXPORTING
            NAME                  = LV_COOKIE_NAME
            APPLICATION_NAMESPACE = RUNTIME->APPLICATION_NAMESPACE
            APPLICATION_NAME      = RUNTIME->APPLICATION_NAME
            USERNAME              = SY-UNAME
```

```
            SESSION_ID            = RUNTIME->SESSION_ID
            DATA_NAME             = I_NAME
          CHANGING
            DATA_VALUE            = E_VALUE.

        E_DURUM-DURUM = ZUTIL_CL_DEFS=>C_DURUM_SUCCESS.

      CATCH CX_ROOT INTO EXREF.
        E_DURUM-DURUM = ZUTIL_CL_DEFS=>C_DURUM_ERROR.
        E_DURUM-DURUM_TEXT = EXREF->GET_TEXT( ).
    ENDTRY.
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZBSP_CL_BASE_CONTROLLER->SET_SESSION_VAR
* +--------------------------------------------------------------------
* | [--->] I_NAME                         TYPE        ZBSP_E_STR
* | [--->] I_VALUE                        TYPE        ANY
* | [<---] E_DURUM                        TYPE        ZUTIL_S_DURUM
* +--------------------------------------------------------------------
  METHOD SET_SESSION_VAR.
    DATA: EXREF TYPE REF TO CX_ROOT.
    TRY .
        RUNTIME->SERVER->SET_SESSION_STATEFUL(  ).


        DATA: LV_DATE TYPE DATUM,
              LV_TIME TYPE UZEIT.

        LV_DATE = SY-DATUM + 1.
        LV_TIME = SY-UZEIT.

        DATA: LV_COOKIE_NAME TYPE ZBSP_E_STR.
        LV_COOKIE_NAME = RUNTIME->APPLICATION_NAME && I_NAME.

        CL_BSP_SERVER_SIDE_COOKIE=>SET_SERVER_COOKIE(
          EXPORTING
            NAME                  = LV_COOKIE_NAME
            APPLICATION_NAMESPACE = RUNTIME->APPLICATION_NAMESPACE
            APPLICATION_NAME      = RUNTIME->APPLICATION_NAME
            USERNAME              = SY-UNAME
            SESSION_ID            = RUNTIME->SESSION_ID
```

```
            DATA_NAME              = I_NAME
            DATA_VALUE             = I_VALUE
          ).

          E_DURUM-DURUM = ZUTIL_CL_DEFS=>C_DURUM_SUCCESS.

        CATCH CX_ROOT INTO EXREF.
          E_DURUM-DURUM = ZUTIL_CL_DEFS=>C_DURUM_ERROR.
          E_DURUM-DURUM_TEXT = EXREF->GET_TEXT( ).
      ENDTRY.
    ENDMETHOD.
ENDCLASS.
```

We are converting a table to json and returning G_JSON in Web Api Page : **ZPPM_BSP_CO_JSON_VIEW**

```
CLASS ZPPM_CL_BSP_CO_JSON_MO DEFINITION
  PUBLIC
  INHERITING FROM ZBSP_CL_BASE_REST_MODEL
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.

    "We will store return value in different formats
    "op request parameter can be used to determine output type
    DATA :G_JSON          TYPE STRING,
          G_JSON_XML      TYPE STRING,
          G_XML           TYPE STRING.

    "Dummy method
    METHODS GET_DATA.

  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.



CLASS ZPPM_CL_BSP_CO_JSON_MO IMPLEMENTATION.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_JSON_MO->GET_DATA
* +-------------------------------------------------------------------
* +-------------------------------------------------------------------
  METHOD GET_DATA.
    "https://help.sap.com/doc/abapdocu_740_index_htm/7.40/en-US/index
    "Creates a dummy table and fills global variables that contains 3
    "Instead you call a different object and retrieve data.

    "Structure for dummy data
    TYPES: BEGIN OF STRUCT,
             DAY  TYPE STRING,
             DATE TYPE D,
           END OF STRUCT.

    "Dummy data
```

```
DATA ITAB TYPE STANDARD TABLE OF STRUCT.
ITAB = VALUE #( ( DAY = 'Yesterday' DATE = SY-DATLO - 1 )
                ( DAY = 'Today'     DATE = SY-DATLO )
                ( DAY = 'Tomorrow'  DATE = SY-DATLO + 1 )
                ).

"Transformation to JSON
DATA(WRITER) = CL_SXML_STRING_WRITER=>CREATE( TYPE = IF_SXML=>CO_
CALL TRANSFORMATION ID SOURCE ITAB = ITAB RESULT XML WRITER.
DATA(JSON) = WRITER->GET_OUTPUT( ).

CALL FUNCTION 'ECATT_CONV_XSTRING_TO_STRING'
  EXPORTING
    IM_XSTRING = JSON
  IMPORTING
    EX_STRING  = G_JSON.


"JSON-XML
DATA(READER) = CL_SXML_STRING_READER=>CREATE( JSON ).
DATA(XML_WRITER) = CL_SXML_STRING_WRITER=>CREATE( ).
READER->NEXT_NODE( ).
READER->SKIP_NODE( XML_WRITER ).

DATA(XML) = XML_WRITER->GET_OUTPUT( ).
CALL FUNCTION 'ECATT_CONV_XSTRING_TO_STRING'
  EXPORTING
    IM_XSTRING = XML
  IMPORTING
    EX_STRING  = G_JSON_XML.


"asXML
CALL TRANSFORMATION ID SOURCE ITAB = ITAB RESULT XML XML.
CALL FUNCTION 'ECATT_CONV_XSTRING_TO_STRING'
  EXPORTING
    IM_XSTRING = XML
  IMPORTING
    EX_STRING  = G_XML.
```

```
  ENDMETHOD.
ENDCLASS.
```

```
CLASS ZPPM_CL_BSP_CO_INDEX DEFINITION
  PUBLIC
  INHERITING FROM ZBSP_CL_BASE_CONTROLLER
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.

    "Page Data Model
    DATA P_MODEL TYPE REF TO ZPPM_CL_BSP_CO_BASE_MO .     "Class that

    "Initialize Class
    METHODS DO_INIT
        REDEFINITION .
    METHODS DO_REQUEST
        REDEFINITION .
  PROTECTED SECTION.

    "Creates Page Model BO
    METHODS CREATE_PAGE_MODEL .

    "Set contoller params
    METHODS SET_CONTROLLER_PARAMS REDEFINITION .

  PRIVATE SECTION.
    DATA: GT_REQ_PRMS TYPE TIHTTPNVP.

    "Refresh Page Data
    METHODS REFRESH_PAGE_DATA.

    "Process Request
    METHODS PROCESS_REQUEST.

    "View Page
    METHODS VIEW_PAGE.

ENDCLASS.



CLASS ZPPM_CL_BSP_CO_INDEX IMPLEMENTATION.
```

```
* <SIGNATURE>---------------------------------------------------------
* | Instance Protected Method ZPPM_CL_BSP_CO_INDEX->CREATE_PAGE_MODEL
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD CREATE_PAGE_MODEL.

*    "Create pages model class
*    ME->P_MODEL ?= ME->CREATE_MODEL(
*      CLASS_NAME = C_PAGE_MODEL_CLS
*      MODEL_ID   = C_PAGE_MODEL_ID ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_INDEX->DO_INIT
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD DO_INIT.
    CALL METHOD SUPER->DO_INIT.

    SET_CONTROLLER_PARAMS( ).
    CREATE_PAGE_MODEL( ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_INDEX->DO_REQUEST
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD DO_REQUEST.
    CALL METHOD SUPER->DO_REQUEST.

    REFRESH GT_REQ_PRMS.
    RUNTIME->SERVER->REQUEST->GET_FORM_FIELDS( CHANGING FIELDS = GT_R

    PROCESS_REQUEST( ).

    VIEW_PAGE( ).
  ENDMETHOD.
```

```
* <SIGNATURE>---------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_INDEX->PROCESS_REQUEST
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD PROCESS_REQUEST.
    "Process Request
    REFRESH_PAGE_DATA( ).
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_INDEX->REFRESH_PAGE_DATA
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD REFRESH_PAGE_DATA.
    "Get Data
    "ME->P_MODEL->GET_DATA( ).

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Protected Method ZPPM_CL_BSP_CO_INDEX->SET_CONTROLLER_PA
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD SET_CONTROLLER_PARAMS.

    C_PAGE                    = 'index.htm'.
    C_PAGE_MODEL_CLS          = 'ZPPM_CL_BSP_CO_JSON_MO'.
    C_PAGE_MODEL_ID           = 'ZPPM_CL_BSP_CO_JSON_MO'.
    C_PAGE_MODEL_VAR          = 'P_MODEL'.
    C_PAGE_CONTROLLER_NAME_VAR = 'P_CONTROLLER_NAME'.

  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_INDEX->VIEW_PAGE
* +---------------------------------------------------------------------
* +---------------------------------------------------------------------
  METHOD VIEW_PAGE.
    "View Page
```

```
    DATA: LV_VIEW TYPE REF TO IF_BSP_PAGE.
    LV_VIEW = CREATE_VIEW( VIEW_NAME = C_PAGE ).

    LV_VIEW->SET_ATTRIBUTE( NAME  = C_PAGE_MODEL_VAR VALUE = ME->P_MC
    LV_VIEW->SET_ATTRIBUTE( NAME  = C_PAGE_CONTROLLER_NAME_VAR VALUE

    CALL_VIEW( LV_VIEW ).
  ENDMETHOD.
ENDCLASS.
```

With this simple Vue and axios example, we have seen how to create a rest based web api with BSP and consume it. In that way, we can create powerful, fast and elegant web applications on SAP. I hope that post inspires many of us who are trying to create powerful, flexible and modern web applications on SAP. Thanks for reading.

Tags:

axios | Fiori Alternative | OData Alternative | vue.js

Comments

larshp Active Contributor

2019 Feb 05
6:35 AM

Cool, suggest sharing the full example on GitHub, as there are multiple parts that has to fit together.

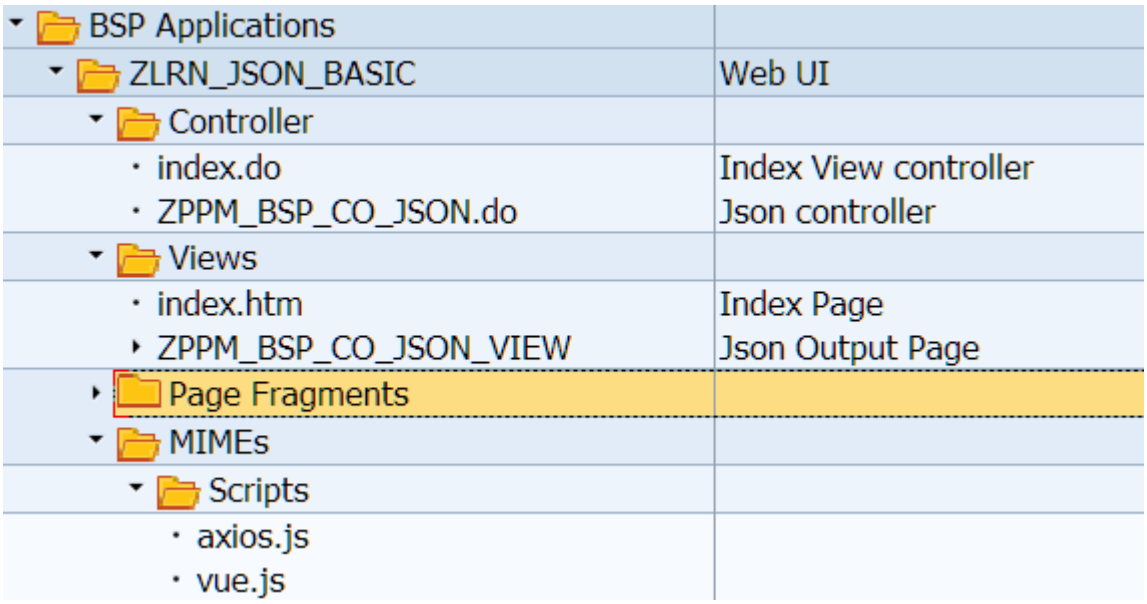
beyhan_meyrali Active Contributor

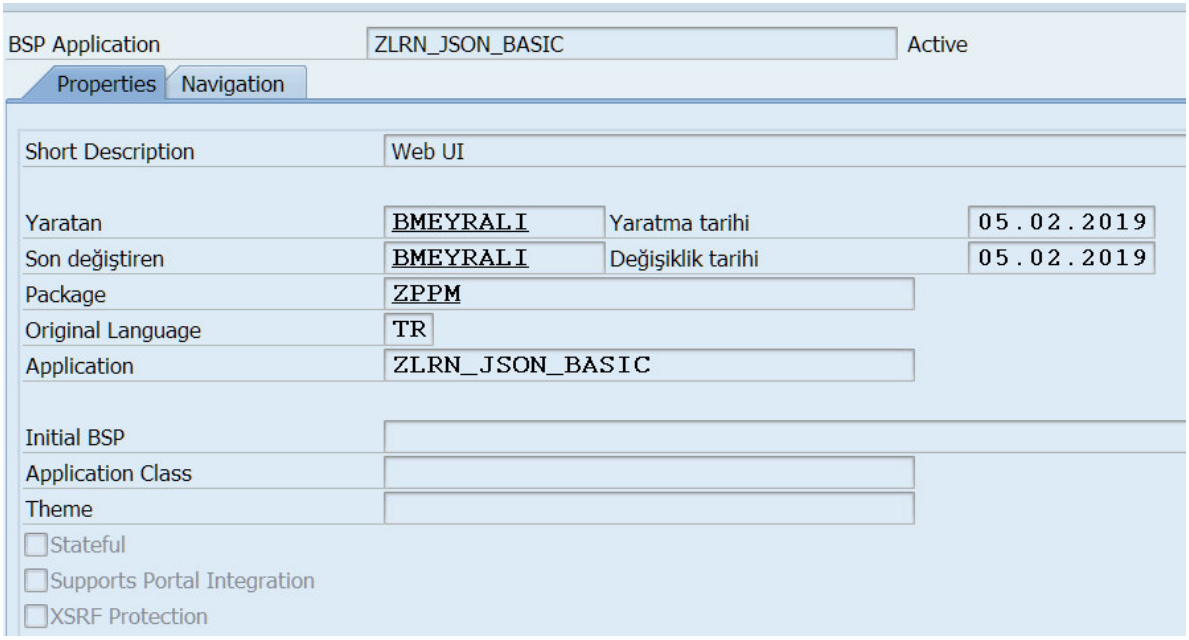
2019 Feb 05
7:20 AM

Hi Lars,

I have created very basic example without my BSP framework. So that should be quite direct to apply.

Structure





Index.do controller class : **ZPPM_CL_BSP_CO_INDEX,**

Does nothing except setting view and displaying...

```
CLASS ZPPM_CL_BSP_CO_INDEX DEFINITION
  PUBLIC
  INHERITING FROM CL_BSP_CONTROLLER2
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.

    "Initialize Class
    METHODS DO_INIT
        REDEFINITION .
    METHODS DO_REQUEST
        REDEFINITION .

  PROTECTED SECTION.
  PRIVATE SECTION.

    METHODS VIEW_PAGE.
ENDCLASS.



CLASS ZPPM_CL_BSP_CO_INDEX IMPLEMENTATION.


* <SIGNATURE>---------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_INDEX->DO_INIT
* +-------------------------------------------------------------
* +-------------------------------------------------------------
  METHOD DO_INIT.
    CALL METHOD SUPER->DO_INIT.
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_INDEX->DO_REQUEST
* +-------------------------------------------------------------
* +-------------------------------------------------------------
  METHOD DO_REQUEST.
    CALL METHOD SUPER->DO_REQUEST.

    VIEW_PAGE( ).
```

```
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_INDEX->VIEW_PAGE
* +-------------------------------------------------------------
* +-------------------------------------------------------------
  METHOD VIEW_PAGE.
    "View Page
    DATA: LV_VIEW TYPE REF TO IF_BSP_PAGE.
    LV_VIEW = CREATE_VIEW( VIEW_NAME = 'index.htm' ).

    CALL_VIEW( LV_VIEW ).
  ENDMETHOD.
ENDCLASS.
```

ZPPM_BSP_CO_JSON.do controller class: **ZPPM_CL_BSP_CO_JSON**

Creates a dummy json and returns in G_JSON

```
CLASS ZPPM_CL_BSP_CO_JSON DEFINITION
PUBLIC
  "Kind of framework to be used in feature BSP pages
  INHERITING FROM CL_BSP_CONTROLLER2
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.

    DATA G_JSON TYPE STRING .

    "Initialize Class
    METHODS DO_INIT REDEFINITION .
    METHODS DO_REQUEST REDEFINITION .

  PROTECTED SECTION.
  PRIVATE SECTION.

    "Create dummy json data from a table
    METHODS GET_DATA.

    "View Page
    METHODS VIEW_PAGE.

ENDCLASS.



CLASS ZPPM_CL_BSP_CO_JSON IMPLEMENTATION.


* <SIGNATURE>---------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_JSON->DO_INIT
* +-------------------------------------------------------------
* +-------------------------------------------------------------
  METHOD DO_INIT.
    CALL METHOD SUPER->DO_INIT.
  ENDMETHOD.


* <SIGNATURE>---------------------------------------------------
* | Instance Public Method ZPPM_CL_BSP_CO_JSON->DO_REQUEST
```

```
* +--------------------------------------------------------------------+
* +--------------------------------------------------------------------+
  METHOD DO_REQUEST.
    CALL METHOD SUPER->DO_REQUEST.

    "Please implement your controller logic here.
    "I suggest to pass params to backend and just retrieve result
    GET_DATA( ).

    VIEW_PAGE( ).
  ENDMETHOD.


* <SIGNATURE>--------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_JSON->GET_DATA
* +--------------------------------------------------------------------
* +--------------------------------------------------------------------
  METHOD GET_DATA.
    "https://help.sap.com/doc/abapdocu_740_index_htm/7.40/en-US/i
    "Creates a dummy table and fills global variables that contai
    "Instead you call a different object and retrieve data.

    "Structure for dummy data
    TYPES: BEGIN OF STRUCT,
             DAY  TYPE STRING,
             DATE TYPE D,
           END OF STRUCT.

    "Dummy data
    DATA ITAB TYPE STANDARD TABLE OF STRUCT.
    ITAB = VALUE #( ( DAY = 'Yesterday' DATE = SY-DATLO - 1 )
                    ( DAY = 'Today'     DATE = SY-DATLO )
                    ( DAY = 'Tomorrow'  DATE = SY-DATLO + 1 )
                    ).

    "Transformation to JSON
    DATA(WRITER) = CL_SXML_STRING_WRITER=>CREATE( TYPE = IF_SXML=
    CALL TRANSFORMATION ID SOURCE ITAB = ITAB RESULT XML WRITER.
    DATA(JSON) = WRITER->GET_OUTPUT( ).

    CLEAR G_JSON.
    CALL FUNCTION 'ECATT_CONV_XSTRING_TO_STRING'
      EXPORTING
```

```
        IM_XSTRING = JSON
      IMPORTING
        EX_STRING  = G_JSON.

  ENDMETHOD.


* <SIGNATURE>-------------------------------------------------------
* | Instance Private Method ZPPM_CL_BSP_CO_JSON->VIEW_PAGE
* +-------------------------------------------------------------------
* +-------------------------------------------------------------------
  METHOD VIEW_PAGE.
    "View Page
    DATA: LV_VIEW TYPE REF TO IF_BSP_PAGE.
    LV_VIEW = CREATE_VIEW( VIEW_NAME = 'ZPPM_BSP_CO_JSON_VIEW' ).

    "We will pass json data
    LV_VIEW->SET_ATTRIBUTE( NAME  = 'G_JSON' VALUE = G_JSON ).
    CALL_VIEW( LV_VIEW ).
  ENDMETHOD.
ENDCLASS.
```

ZPPM_BSP_CO_JSON_VIEW view page

Has only one attribute G_JSOn type string

Code:

```
<%@page language="ABAP" %>
<%= G_JSON %>
```

index.htm to query json controller with axios and display with Vue.

No attributes

```
<HTML>
<HEAD>
  <!-- production version, optimized for size and speed -->
  <script src="Scripts/vue.js"></script>
  <script src="Scripts/axios.js"></script>
</HEAD>

<body>

  <div id="app2">
    <div v-if="loading">
      loading...
    </div>
    <div v-else>
      <ul>
        <li v-for="item in items.ITAB">
          <h1>
            {{item.DAY}}
          </h1>
          <p>
            {{item.DATE}}
          </p>
        </li>
      </ul>
    </div>
  </div>


  <h1>Test Page</h1>
  <p><a href="ZPPM_BSP_CO_JSON.do">TEST K#sa</a></p>

  <script>

    const URL = 'ZPPM_BSP_CO_JSON.do';
    var app2 = new Vue({
      el: '#app2',
      data() {
        return {
          loading: true,
      posts: [] // add posts here so reactivity is working, also
    }
  },
```

```
  created() {
    //this.loading = true --> not needed already set in data
    //op:03 represents Json format, see base class constants: ZBS
    axios.post(URL, {
      params: {
        op: '03'
      }}).then((response) => {
      // console.log(response.data, this)
      this.items = response.data
      this.loading = false
    })
    }
  })
</script>
</body>

</HTML>
```

That is all. Any questions, please write.

Thanks for comment.

mmcisme1 Active Contributor

2019 Feb 06
1:09 PM

So very cool! We have web developers who are learning everything. This will give them a better comfort zone. IE Quicker turn around while learning. I'll share it with them.

beyhan_meyrali Active Contributor

2019 Feb 11
8:20 AM

Hi,

Thanks for comment :).

Here is an useful example. In fact, you can copy and create apps with that sample.

```
<!DOCTYPE html>
<html>
<head>
    <link href="https://fonts.googleapis.com/css?family=Roboto:10
    <link href="https://cdn.jsdelivr.net/npm/vuetify/dist/vuetify
    <meta name="viewport" content="width=device-width, initial-sc

    <script src="https://npmcdn.com/vue/dist/vue.js"></script>
    <script src="https://npmcdn.com/vue-router/dist/vue-router.js
    <script src="https://cdn.jsdelivr.net/npm/vuetify/dist/vuetif
    <script src="https://unpkg.com/axios/dist/axios.min.js"></scr
</head>
<body>
    <div id="app">
        <v-app id="inspire">
            <v-toolbar>
                <v-toolbar-title>Toolbar Mobile Menu</v-toolbar-t
                <v-spacer></v-spacer>
                <v-toolbar-items class="hidden-sm-and-down">
                    <v-btn
                    v-for="item in menu"
                    :key="item.key"
                    :to="item.link"
                    flat
                    >{{ item.title }}</v-btn>
                </v-toolbar-items>

                <v-menu bottom left class="hidden-md-and-up">
                    <v-btn slot="activator" icon>
                        <v-icon>menu</v-icon>
                    </v-btn>
                    <v-list>
                        <v-list-tile v-for="item in menu" :key="i
                            <v-list-tile-content>
                                <v-list-tile-title>{{ item.title
                            </v-list-tile-content>
                        </v-list-tile>
                    </v-list>
                </v-menu>
            </v-toolbar>
            <v-content>
                <v-container fluid fill-height>
```

```
                <v-layout justify-center align-center>
                    <v-flex shrink>
                        <router-view></router-view>
                    </v-flex>
                </v-layout>
            </v-container>
        </v-content>
    </v-app>

</div>


<template id="home-template">
    <div>Home</div>
</template>


<template id="foo-template">
    <div>Foo</div>
</template>


<template id="nf-template">
    <div>Not Found!</div>
</template>


<template id="projects-template">

    <div class="post">
        <div v-if="loading" class="loading">
            Loading...
        </div>

        <div v-else-if="error" class="error">
            {{ error }}
        </div>

        <div v-else class="content">
            <ul>
                <li v-for="proj in projects">
                    <h1>
                        {{proj.DAY}}
```

```
                    </h1>
                    <p>
                        {{proj.DATE}}
                    </p>
                </li>
            </ul>
        </div>

    </div>

</template>


<script>
    //Page to query json data
    const URL = '/sap(bD1lbiZjPTEwMA==)/bc/bsp/sap/zppm_web_u


    //Page components
    const Home = Vue.component('home', {    template: '#home-
    const Foo = Vue.component('foo', {  template: '#foo-templ
    const Notfound = Vue.component('nf', {  template: '#nf-te

    //Project component
    const Projects = Vue.component('projects'
        , { template: '#projects-template'
        , data () {
            return {
                loading: false,
                error: null,
                projects: null,
            }
        }
        , mounted () {
            // fetch the data when the view is created and th
            // already being observed
            this.getCat2Data()
        }

        , methods: {
            getCat2Data () {
                this.error = null
                this.loading = true
```

```
            //Place axios calls here
            //op:03 represents Json format, see base clas
            axios.post(URL, {
                params: {
                    op: '03'
                }
            }).then(
                (response) => {
                    this.projects = response.data.ITAB
                    this.loading = false },
                (error) => {
                    console.log(error)
                    this.loading = false
                    this.error = "Could not reach the API
                }
            );
        }
    }
}
);

//Router
const router = new VueRouter({
    mode: 'history',
    routes: [
    { path: '/', component: Home },
    { path: '/foo', component: Foo },
    { path: '*', component: Notfound },

    { path: '/projects', component: Projects }
    ]
})


//Menu elements
const menu = [
{ key: '1', title: 'Link A', link:'/' },
{ key: '2', title: 'Link B', link:'/foo' },
{ key: '3', title: 'Link C', link:'/about' },

{ key: '4', title: 'Projects', link:'/projects' }
]
```

```
        //Vue app
        const mainApp = new Vue({
            router,
            el: '#app',

            data () {
                return {
                    menu
                }
            }
        })

    </script>
</body>
</html>
```

mmcisme1 Active Contributor

2019 Feb 11
12:38 PM

Even better an example.

Thank you!